

From Observation to Action: Latent Action-based Primitive Segmentation for VLA Pre-training in Industrial Settings

Jiajie Zhang Sören Schwertfeger
School of Information Science and Technology
ShanghaiTech University, Shanghai, China
{zhangjj2023, soerensch}@shanghaitech.edu.cn

Alexander Kleiner
School of Automation
Hangzhou Dianzi University, China
alexander.kleiner@gmail.com

Abstract

We present a novel unsupervised framework to unlock vast unlabeled human demonstration data from continuous industrial video streams for Vision-Language-Action (VLA) model pre-training. Our method first trains a lightweight motion tokenizer to encode motion dynamics, then employs an unsupervised action segmenter leveraging a novel "Latent Action Energy" metric to discover and segment semantically coherent action primitives. The pipeline outputs both segmented video clips and their corresponding latent action sequences, providing structured data directly suitable for VLA pre-training. Evaluations on public benchmarks and a proprietary electric motor assembly dataset demonstrate effective segmentation of key tasks performed by humans at workstations. Further clustering and quantitative assessment via a Vision-Language Model confirm the semantic coherence of the discovered action primitives. To our knowledge, this is the first fully automated end-to-end system for extracting and organizing VLA pre-training data from unstructured industrial videos, offering a scalable solution for embodied AI integration in manufacturing.

1. Introduction

The pursuit of developing generalist agents, frequently embodied as Vision-Language-Action (VLA) models, constitutes a major objective within the robotics research community [1, 2, 31]. Pre-training these models on large-scale, diverse datasets, encompassing human video recordings [15] and robot trajectories from multiple embodiments [23], has proven crucial for enabling strong generalization and reliable instruction-following performance. Applying this approach for real-world deployment encounters a fundamental problem: the scarcity of training data. In contrast to the vast amounts of text and images available on the Internet, obtaining high-quality action-annotated robot data remains challenging and costly and typically necessitating expensive

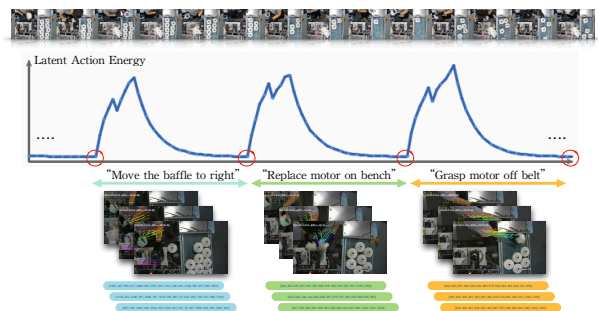


Figure 1. Example of our segmentation approach using *Latent Action Energy* from a *Motion Tokenizer*. Action boundaries (red circles) correspond to transitions from high energy to baseline, indicating action completion. The pipeline outputs the *Latent Action Sequence* (bottom codes), providing structured representations for VLA pre-training.

teleoperation [5, 17].

Our work focuses on the manufacturing domain, specifically on industrial environments characterized by constrained operational spaces where meaningful human actions comprise a finite, well-defined set. We introduce a fully unsupervised system that is able to extract a core "vocabulary" of actions through passive observation. Many state-of-the-art VLA pretraining strategies [1, 5] adopt a hierarchical framework: (1) a high-level generalist module that encodes arbitrary action sequences into abstract latent action tokens, and (2) a low-level controller for task execution that is subsequently fine-tuned through supervised learning on these token sequences [9, 29]. Our work focuses on training the high-level generalist module, which shifts the data bottleneck earlier by requiring large collections of pre-segmented video clips paired with latent token sequences from curated datasets [15]. We are tackling the key challenge on how to automatically extract structured data from vast, unstructured video streams found online and in industrial settings.

We are presenting a solution that allows robots to learn

like humans through continuous, self-directed observation of actions and behaviors [2]. To this end, we introduce a novel unsupervised framework for automatically discovering and segmenting action primitives from continuous video streams. Our method first trains a lightweight motion tokenizer to encode motion dynamics, then employs an unsupervised action segmenter leveraging a novel metric named *Latent Action Energy* to discover and segment semantically coherent action primitives. The pipeline outputs both segmented video clips and their corresponding latent action sequences, providing structured data directly suitable for VLA pre-training (see Figure 1). Our main contributions are summarized as follows:

- We introduce a novel segmentation approach based on *Latent Action Energy*, a metric defined in the abstract latent action space, to identify semantic action primitives from raw video data. This differs from conventional methods that focus on pixel-level or optical flow changes [12].
- We present an *end-to-end automated data pipeline* that transforms hours-long industrial video footage into a structured repository of action primitives. This directly addresses the data sourcing bottleneck for industrial Vision-Language-Action (VLA) latent pretraining.
- We are the first to validate this VLA data-sourcing methodology on publicly available benchmark datasets and a genuine and complex industrial data set from an assembly line. Section 4 provides strong quantitative and qualitative evidence demonstrating its practical feasibility and scalability.

To foster reproducibility and visually demonstrate our pipeline’s effectiveness, our code and proprietary industrial dataset will be made publicly available upon acceptance, supported by an accompanying video in the supplementary material.

2. Related Work

Our research is positioned at the intersection of generalist robot policies, latent action representation, and unsupervised action segmentation from video, with a specific focus on industrial applications. In this section, we review the state-of-the-art in these areas to highlight the gap our work aims to fill.

Generalist Robot Policies. The development of generalist robot policies is based on Vision-Language-Action (VLA) models [4, 18, 31], which are large-scale foundation models trained on web-scale data combined with action modalities for robot control. Notable examples include GR00T [1] and AgiBot GO-1 [5], both of which have recently made significant advances. To ensure that these models generalize over various tasks, they must be trained on huge amounts of heterogeneous data [15, 23]. However, this approach encounters a significant bottleneck since it

requires a large dataset of *pre-segmented, action-labeled* video clips, which are usually obtained through expensive teleoperation [5, 17].

Latent Action Representation from Video Data. A key strategy to overcome the labeled data bottleneck is to learn action priors from video data by leveraging *latent action representations*, an abstract, embodiment-agnostic space for modeling behaviors [10]. Early pioneering works including LAPO and LAPA have demonstrated the effectiveness of latent action learning [26, 29]. Nevertheless, both approaches depend on pixel-level objectives, including next-frame prediction for simple game actions in LAPO and VQ-VAE-based reconstruction in LAPA. This can lead to capturing action-irrelevant background noise and may provide limited descriptive capacity. Importantly, these successful techniques typically assume access to curated short video clips (e.g., Ego4D [15]), and do not address the upstream challenge of discovering and segmenting action primitives from continuous video streams [1, 5]. Our work is unique in that it capitalizes on the utility of latent action representations while re-purposing them for identifying temporal boundaries of action primitives themselves. As described in the latter sections, this is carried out by utilizing keypoint-based dynamics [16] with a motion tokenizer [10].

Unsupervised Action Segmentation. The task of Temporal Action Detection (TAD), which involves identifying the boundaries between actions, is commonly addressed using local boundary detectors [12]. While fully-supervised [27] and weakly-supervised [28], e.g., transcript-based methods, provide efficient architectures and alignment strategies, our approach operates in a completely unsupervised manner, requiring no labels. Unsupervised techniques such as ABD [13] detect change-points by locating local minima in the similarity of visual features, but they are often sensitive to non-semantic physical changes like lighting variations. More sophisticated methods like OTAS [20] address this issue by combining explicit features, for instance, by using object detectors and Graph Neural Networks (GNNs), which introduce significant complexity. Our method introduces for action primitive discovery a fundamentally different method based on *Latent Action Energy* that is derived from the *latent action space* of a VLA-oriented Motion Tokenizer. Instead of detecting visual similarity valleys or fusing object features, we identify primitives through sustained high-energy activations, implicitly capturing semantic motion without object detectors. This shift from “visual change detection” to “behavioral intent change detection” directly addresses the need for VLA pre-training data.

Robot Learning in Industrial Environments. Prior work on generalist robots has predominantly focused on home and laboratory settings [2, 17] with diverse and unstructured task distributions. Industrial environments, how-

ever, present a distinct and highly valuable domain [3], characterized by structured, repetitive workflows and a finite, countable set of skilled actions. While large-scale efforts like AgiBot World [5] have begun to include industrial scenarios, the data remains primarily sourced through manual teleoperation. Our work specifically targets on this high-impact domain. By leveraging the inherent structured nature of industrial environments, we develop an automated VLA-centric data pipeline that can autonomously discover the complete "action vocabulary" of workstations through passive observation. This approach offers a scalable solution for deploying and continuously enhancing VLA models within real-world manufacturing settings.

3. System Methodology

Our system introduces a fully unsupervised pipeline that transforms continuous, unlabeled industrial video streams into a structured repository of action primitives suitable for VLA pre-training. Figure 2 depicts the LAPS (Latent Action-based Primitive Segmentation) pipeline which processes data through three sequential stages: (1) *Motion Tracking*: Using point trackers such as CoTracker [16], dense motion trajectories are extracted from raw video streams and stored in a sliding window buffer of motion keypoints. (2) *Action Detection & Segmentation*: The keypoints from the sliding window are fed into a motion tokenizer that generates a continuous *Latent Action Vector Stream*. An action detector applies a hysteresis-based controller to this stream using our novel *Latent Action Energy* metric to identify sustained action activations. The primitive segmentor then uses these detected activations to locate action boundaries and to extract action primitives consisting of *Segmented Latent Vectors* along with their corresponding video clips and action codes. (3) *Semantic Action Clustering*: The identified latent vectors are clustered via temporal embedding and k -means to automatically discover the finite set of *Semantic Action Clusters*, thereby determining the complete set of workstation tasks.

3.1. Action Detection & Segmentation

The core of our segmentation approach relies on representing video content within an abstract, high-dimensional latent action space rather than operating directly on raw pixels. To accomplish this, we first train a lightweight motion tokenizer M_θ . Our architecture is primarily derived from the temporal quantized autoencoder introduced in AMPLIFY [10], which comprises a Transformer-based encoder (E_θ) and decoder (D_θ), along with a Finite Scalar Quantization (FSQ) [21] layer for discretization. The tokenizer is trained on a large dataset of short video clips, denoted as $\mathcal{D}_{\text{clips}}$. For each clip, we first extract a dense grid of N keypoint tracks using an off-the-shelf point tracker [16]. These tracks are then consolidated into a single tensor $\kappa \in$

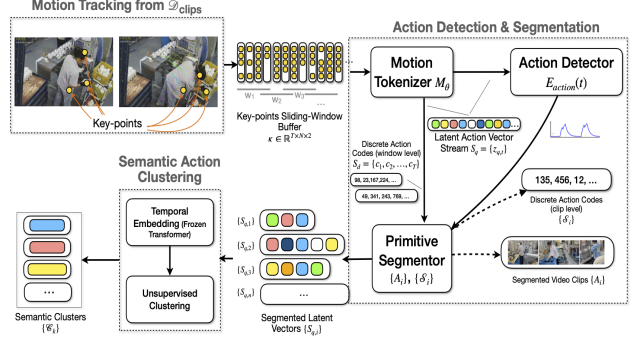


Figure 2. Overview of the LAPS pipeline: (1) *Motion Tracking* extracts motion keypoints from raw video using a point tracker. (2) *Action Detection & Segmentation* generates a latent vector stream via a motion tokenizer and identifies action boundaries to segment latent vectors, video clips, and action codes. (3) *Semantic Action Clustering* groups the segmented latent vectors into meaningful semantic action clusters.

$\mathbb{R}^{T \times N \times 2}$, where T indicates the temporal length (number of frames), N is the number of tracked points, and 2 corresponds to the (x, y) spatial coordinates. The encoder E_θ transforms the velocities derived from these tracks into a latent sequence, which is then discretized into tokens $z_t \in \mathcal{Z}$ using FSQ. Rather than reconstructing pixels, the decoder D_θ is trained with a classification objective to predict the relative displacement of each track point. This is achieved by applying a cross-entropy loss over a discrete spatial grid, effectively modeling motion dynamics as a categorical distribution. A comprehensive description of the tokenizer’s architecture and training methodology is provided in the supplementary material.

While AMPLIFY [10] employed this representation for policy learning, our work adapts it for a novel application: providing the primary signal for unsupervised temporal segmentation. Continuous video streams are processed by using a sliding-window approach, as illustrated in Figure 3. The Motion Tokenizer M_θ generates two complementary representations for each window:

1. A sequence of *continuous quantized vectors* $S_q = \{z_{q,1}, \dots, z_{q,T}\}$, where each $z_{q,t} \in \mathbb{R}^{d_m}$ corresponds to a vector prototype from the FSQ codebook.
2. A sequence of *discrete code indices* $S_d = \{c_1, \dots, c_T\}$, where each c_t is an integer index.

Although the *discrete* code indices from the window-level S_d are the components used to construct the final segment-level sequences S_i , the *continuous* quantized vector sequence S_q is critical for our pipeline’s internal operation. This continuous sequence S_q preserves richer geometric and semantic information than the discrete indices alone.

Consequently, the continuous window-level stream S_q serves as the foundational signal for our segmentation framework, as its continuous nature is required for calcu-

lating our E_{action} metric via temporal differences by the L_2 norm. Furthermore, the segment-level vectors derived from this stream denoted as $S_{q,i}$ serve as the input for our subsequent clustering analysis. This continuous representation enables meaningful distance computations within the latent space.

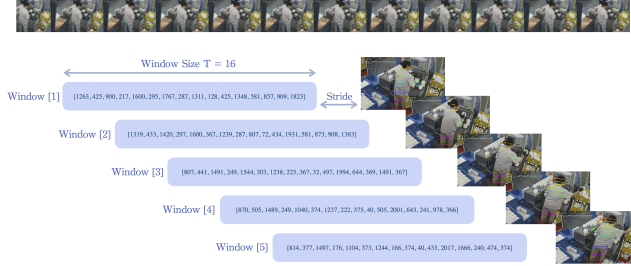


Figure 3. Sliding-window tokenization: A motion tokenizer converts the video stream into a sequence of discrete latent action indices $c_t \in \{0, \dots, 2047\}$, the main output for VLA pre-training. Action detection and clustering use the corresponding continuous quantized vectors.

Traditional unsupervised segmentation methods which are based on low-level metrics such as optical flow[12] or visual similarity [13], capture physical motion. These signals are often volatile and sensitive to pixel-level variations that may not reflect a true change in the task’s semantic phase. In contrast, our approach targets *semantic intent* as the driver of meaningful boundaries (e.g., transitions from “reaching” to “grasping”). Such intent-driven shifts induce distinct systematic changes in motion dynamics. Our *Motion Tokenizer* (Section 3.1) is well-suited to this objective: its abstract Latent Action Sequence S explicitly models these dynamics while suppressing low-level visual noise.

3.1.1. Mathematical Definition of Latent Action Energy

We introduce a novel metric termed *Latent Action Energy* E_{action} . This metric is defined directly on the dynamics of the continuous quantized vectors $S_q = \{z_{q,1}, \dots, z_{q,T}\}$ produced by the motion tokenizer. We formally define the Latent Action Energy as the L_2 norm of the temporal difference within the quantized latent space:

$$E_{\text{action}}(t) = \|z_{q,t} - z_{q,t-1}\|_2$$

This formulation is resilient to appearance changes yet remains acutely responsive to shifts in latent motion dynamics. The energy metric $E_{\text{action}}(t)$ remains low when the latent action token $z_{q,t}$ is stable (e.g., during inactivity), and exhibits sustained high activation when tokens vary dynamically (i.e., throughout a continuous, coherent action primitive). A semantic shift—marking an action boundary—is detected when the energy signal returns to a low state, indicating the conclusion of the preceding action.

3.1.2. Unsupervised Action Detection

The *Action Detector* operates as a causal state machine rather than a conventional peak-detection algorithm. Specifically, it implements a robust two-state (ON/OFF) controller with hysteresis that processes the one-dimensional time-series signal $E_{\text{action}}(t)$. This single-pass architecture enables real-time operation and facilitates online data curation. The segmentation procedure is as follows:

1. *Causal Signal Smoothing*: The raw energy signal $E_{\text{action}}(t)$ is smoothed using an exponential moving average (EMA) to reduce high-frequency noise without future data leakage:

$$y_t = \alpha E_{\text{action}}(t) + (1 - \alpha)y_{t-1}$$

where y_t is the smoothed signal and α the smoothing factor.

2. *Online Boundary Detection with Hysteresis*: A two-state (ON/OFF) controller with hysteresis detects segment boundaries on y_t :

- *Activation (OFF $\rightarrow$ ON)*: Triggered when $y_t > \theta_{\text{on}}$ for u consecutive frames.
- *Deactivation (ON $\rightarrow$ OFF)*: Ends when $y_t < \theta_{\text{off}}$ for d consecutive frames, with $\theta_{\text{off}} \leq \theta_{\text{on}}$.

This dual-threshold and debounce scheme ensures stable segmentation in noisy streaming data.

3. *Primitive and Sequence Extraction*: Upon OFF transition, the segment $A_i = V[p_i : p_{i+1}]$ is extracted from $V \in \mathcal{D}_{\text{clips}}$ corresponding to the active interval. Additionally, the method outputs the corresponding discrete FSQ code indices $\{c_t\}$ overlapping the segment from the sequence $S_i = \{c_{p_i}, \dots, c_{p_{i+1}}\}$, preserving temporal dynamics for VLA pre-training.

The primary threshold θ_{on} is determined through a fully unsupervised offline optimization procedure leveraging self-supervised pseudo-labeling, eliminating the need for manual annotations. The process is as follows:

1. *Proxy Signal and Pseudo-Label Generation*: We first compute a simple, low-level velocity energy from the temporal difference of velocity keypoints over a validation dataset to serve as a proxy signal. A heuristic method for auto thresholding [22] is applied to the proxy signal to automatically generate binary pseudo-labels, y_{pseudo} , which provide a coarse-grained distinction between “motion” and “non-motion” windows.
2. *Threshold Optimization*: With these noisy y_{pseudo} labels as the optimization target, we then perform a parameter sweep to find the optimal θ_{on} for our high-level *Latent Action Energy* signal, $E_{\text{action}}(t)$. We select the θ_{on} that maximizes the $F1$ -score as quality metric between the thresholded $E_{\text{action}}(t)$ and the pseudo-labels y_{pseudo} .

This self-supervised, two-stage procedure enables the robust calibration of the threshold for our highly descriptive

latent-space energy signal by leveraging a simple velocity-based signal. The resulting dataset-level threshold, θ_{on} , is fixed for all subsequent online segmentation tasks. The lower threshold, θ_{off} , is then computed as a fraction of this primary threshold ($\theta_{\text{off}} = r \cdot \theta_{\text{on}}$), where r denotes the hysteresis factor ($0 < r \leq 1$).

3.2. Semantic Action Clustering

Following the segmentation of the continuous video stream into a collection of variable-length action primitives $\mathcal{A} = \{A_1, \dots, A_N\}$ via our action segmentation method, the final step involves the unsupervised discovery of the finite set of actions intrinsic to the workstation. This is formulated as an unsupervised clustering task aimed at validating the semantic consistency of the segmented actions.

Each primitive A_i is represented by its corresponding Latent Action Sequence S_i . As detailed in [10], for clustering we utilize the continuous feature vectors obtained from the Motion Tokenizer’s quantization pipeline, denoted as

$$S_{q,i} = [z_{q,1}, \dots, z_{q,T_i}] \in \mathbb{R}^{T_i \times d_m},$$

where T_i represents the variable sequence length of primitive i , and d_m is the dimensionality of the descriptor (e.g., 768). The objective is to cluster these high-dimensional, variable-length time series into semantically coherent clusters without labels.

3.2.1. Temporal Embedding via Frozen Transformer

To capture the temporal dependencies within each sequence $S_{q,i}$, we utilize a lightweight transformer encoder. Importantly, this model functions entirely in a training-free inference mode: all parameters, including projection layers and self-attention weights, remain at their randomly initialized values and are never updated [30]. This design choice is essential to achieve industrial scalability. This ensures generalization across domains, eliminates the need for manual annotations, minimizes computational requirements, and provides inherent robustness to domain shifts by avoiding overfitting to a particular training corpus. The architecture processes each sequence $S_{q,i}$ as follows:

1. *Projection & Encoding*: Input vectors $z_{q,t}$ are linearly projected into the model dimension d , and sinusoidal positional encodings PE_t are added.
2. *Transformer Encoder*: A stack of L layers with H multi-head self-attention heads processes the token sequence.
3. *Pooling*: The final sequence of hidden states $H^{(L)} \in \mathbb{R}^{T_i \times d}$ is aggregated into a single segment-level embedding $e_i \in \mathbb{R}^d$.

Through systematic hyperparameter search (detailed in the Experiments section), we select mean pooling ($e_i = \frac{1}{T_i} \sum_t h_t^{(L)}$) with $d = 256$, $L = 4$, and $H = 4$. Empirically, in this frozen setting, mean pooling demonstrates superior stability and discriminative power compared to alternatives

such as CLS [8] or attention pooling, which rely on parameters that would otherwise need to be learned.

3.2.2. Action Clustering via Cosine k -means

Clustering high-dimensional embeddings with $d = 256$ presents significant challenges, as vector orientation typically carries more discriminative information than magnitude in such spaces. Consequently, we adopt cosine geometry as the foundation for cluster separation. We employ k -means as our primary clustering algorithm, making it compatible with cosine distance through a two-step preprocessing procedure. First, we standardize all embeddings e_i to zero mean and unit variance, then apply L_2 -normalization to obtain $\hat{e}_i = e_i / \|e_i\|_2$. Applying standard k -means (which minimizes Euclidean distance) on these normalized vectors $\hat{e}_i$ is mathematically equivalent to optimizing cosine similarity, since

$$\|\hat{e}_i - \hat{e}_j\|_2^2 = 2(1 - \cos(\hat{e}_i, \hat{e}_j)).$$

A fundamental hypothesis of our work is that fixed industrial workstations exhibit a finite and countable set of core action primitives. Therefore, k is not treated as a free parameter to be optimized by internal metrics. Instead, k is set a priori based on domain expertise and empirical observation of the workstation’s operational tasks. The k -means algorithm then partitions the segmented primitives $\mathcal{A}$ into k clusters $\{\mathcal{C}_1, \dots, \mathcal{C}_k\}$. The success of our approach depends critically on whether these discovered clusters align with semantically meaningful action categories (e.g., “move the baffle,” “pick up a motor”).

3.2.3. Semantic Validation via Vision-Language Model

To quantitatively evaluate the semantic coherence of the discovered clusters, a task for which conventional internal metrics like the Silhouette score are insufficient due to their exclusive focus on spatial separation, a pre-trained Vision-Language Model (VLM) [24] is deployed. We propose a metric, *Intra-Cluster Semantic Similarity (ICSS)*, defined as follows:

1. *Segment Embedding*: For each video primitive A_i , we compute a fixed-length embedding v_i . This is achieved by sampling multiple frames from A_i , extracting their ℓ_2 -normalized visual features using the VLM’s encoder (e.g., CLIP ViT-B/32), and aggregating them via a *norm-weighted pooling* strategy. This produces a final, ℓ_2 -normalized segment descriptor v_i that captures the holistic visual content.
2. *Pairwise Similarity*: For each cluster $\mathcal{C}_k$, we sample a large set of primitive pairs $\mathcal{P}_k = \{(A_i, A_j) \mid A_i, A_j \in \mathcal{C}_k, i \neq j\}$.
3. *Metric Calculation*: The similarity for that cluster is the

average cosine similarity of all sampled pairs:

$$\text{ICSS}_k = \frac{1}{|\mathcal{P}_k|} \sum_{(i,j) \in \mathcal{P}_k} \cos(v_i, v_j)$$

A high average ICSS score, especially when compared to a *random-pair baseline* (computed by sampling the same number of pairs from the *entire* dataset irrespective of clusters), indicates that our pipeline successfully groups semantically meaningful and similar actions. This validation serves a dual purpose: it confirms the precision of our action segmentor and the descriptive power of the temporal embedding (Section 3.2.1).

4. EXPERIMENTS

To validate our proposed framework, we conduct a series of experiments designed to answer three key questions: (1) Does the proposed *Latent Action Energy* provide a more effective signal for semantic action segmentation compared to traditional metrics? (2) How does our unsupervised action segmentor compare with state-of-the-art unsupervised temporal action detection (TAD) baselines? (3) Do the segmented action primitives form semantically coherent and finite clusters confirming their quality for VLA pre-training?

4.1. Experimental Setup

For our experiments, the following datasets have been used: **GTEA** [14] is a dataset comprising 28 videos with a combined duration of approximately 35 minutes, recorded by 4 participants performing tasks in a single kitchen environment. The dataset encompasses 7 procedural activities, each averaging 1.5 minutes in length, captured using a head-mounted camera system. **Breakfast** [19] consists of 1,712 videos documenting 10 distinct cooking activities. This dataset presents several challenges, including substantial temporal variation in video length (ranging from 30 seconds to 7 minutes), frequent occlusions, and multiple camera viewpoints. **Industrial Motor Assembly Dataset** is a new, self-collected dataset from a real-world electro motor assembly line, containing ~ 10 hours of continuous videos from two synchronized views (top-down and exocentric). A two hour long test subset was created and annotated for quantitative comparison against traditional Temporal Action Detection (TAD) baselines. This annotated subset will be made publicly available to promote reproducibility and support future research.

4.1.1. Implementation Details

We implemented the pipeline following the approach detailed in Section 3. The implementation details are as follows: 1) The *Motion Tokenizer* (M_θ) [10] was trained exclusively on unlabeled clips drawn from the training partition of our dataset. 2) The parameters of the *Action Segmentor*

(including $\theta_{\text{on}}, \theta_{\text{off}}, u, d$) were tuned via the unsupervised calibration procedure described in Section 3.1.2. 3) For *Clustering* (Section 3.2), latent sequences $S_{q,i} \in \mathbb{R}^{T_i \times 768}$ were embedded using a frozen Transformer model with $L = 4$ layers and $H = 4$ heads, followed by mean pooling to obtain embeddings $e_i \in \mathbb{R}^{256}$. This parameter-efficient encoder, containing approximately 2.3 million parameters without requiring pre-training, provides a computationally lightweight solution well-suited for large-scale deployment scenarios. Finally, we applied cosine k -means clustering (Section 3.2.2) with a predefined number of clusters k determined based on domain knowledge and empirical observation.

4.1.2. Baselines and Metrics

We evaluate the LAPS pipeline against the following three representative unsupervised action segmentation approaches:

- *Optical Flow Baseline*: Instantiates the traditional *physical motion* paradigm. We apply the identical online state-machine architecture (Section 3.1.1) to standard *Optical Flow Magnitude* features, enabling direct and controlled comparison of signal quality relative to our proposed E_{action} signal (Section 4.2).
- *ABD* [13]: A state-of-the-art method exemplifying the *local boundary detection* paradigm, which identifies temporal change-points through local minima in visual feature similarity.
- *OTAS* [20]: A state-of-the-art method representing the *explicit feature-fusion* paradigm, integrating global, object-interaction, and object-relation features for boundary detection.

Evaluation Metrics: Temporal segmentation accuracy is measured using strict boundary-level *F1-scores* with 2 second and 5 second tolerances (F1@2s, F1@5s), consistent with [20]. Clustering quality (Section 4.4) is assessed via complementary unsupervised metrics: Silhouette Score [25], Calinski-Harabasz Index [6], and our proposed *Intra-Cluster Semantic Similarity (ICSS)* metric.

4.2. Effectiveness of the Latent Motion Energy

To validate our central hypothesis that segmentation is most effectively performed within the latent semantic space, we begin by evaluating the segmentation signal E_{action} . The qualitative result, illustrated in Figure 4, confirms our hypothesis that the latent action space provides a more discriminative representation for semantic boundary detection compared to pixel-space and optical flow-based approaches. It effectively captures transitions in task-phase intent while suppressing the influence of spurious physical motion artifacts.

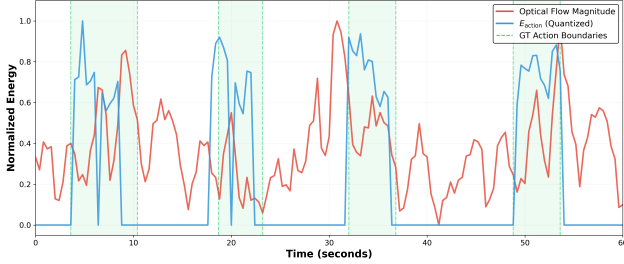


Figure 4. Qualitative comparison of our E_{action} (blue) and Optical Flow (red) over 60 seconds. Our latent action energy shows clear, sustained peaks during actions and sharp drops at ground truth semantic boundaries (dashed lines), while optical flow is noisy and reflects only physical movement, not task phases.

4.3. Unsupervised Action Segmentation

Quantitative comparisons against unsupervised TAD baselines are presented in Tables 1 and 2. On public benchmarks (Table 1), LAPS achieves performance comparable to state-of-the-art approaches despite using only lightweight training (approx. 25 mins) on raw video, contrasting with baselines reliant on extensive pre-training (e.g., I3D [7]). Crucially, on the industrial dataset (Table 2), LAPS demonstrates superior performance. While conventional methods like ABD [13] degrade due to sensitivity to physical motion variations in low-level features [11], our *Motion Tokenizer* robustly captures semantic transitions. This capability is evidenced by high F1@2s scores, underscoring our precision in segmenting repetitive, countable actions inherent to industrial workflows.

Table 1. Comparison on GTEA [14] and Breakfast [19].

Method	GTEA		Breakfast	
	F1@5s	F1@2s	F1@5s	F1@2s
ABD [13]	81.92	74.23	54.50	33.33
OTAS [20]	37.68	36.90	62.13	39.49
LAPS (Ours)	73.12	63.20	58.82	36.72

Table 2. Comparison on the Industrial Motor Assembly.

Method	Top-down View		Exocentric View	
	F1@5s	F1@2s	F1@5s	F1@2s
Optical Flow	56.96	43.68	66.06	42.54
ABD [13]	53.00	34.08	50.32	29.86
OTAS [20]	62.24	40.69	54.56	33.38
LAPS (Ours)	84.26	81.27	84.75	81.93

4.4. Quality of Action Primitives

One of our central claims is that our segmented primitives are besides being well-bounded are also semanti-

cally meaningful, thereby constituting a discrete and countable set of actions. We validate this claim by applying our clustering pipeline (*Frozen Transformer* + *k-means*) to all primitives segmented from the training set. Fig-

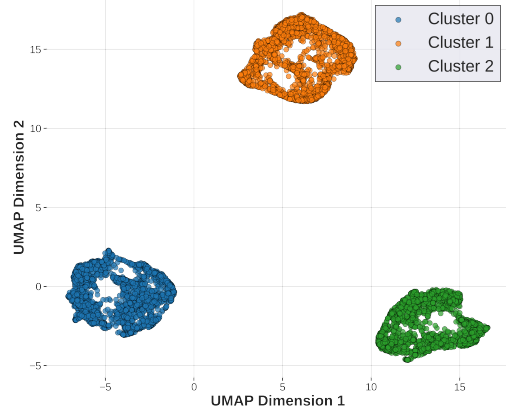


Figure 5. UMAP visualization of action primitive embeddings colored by k-means cluster ID. Distinct, well-separated clusters that correspond to real workstation tasks confirmed through manual inspection.

ure 5 presents a UMAP visualization of the action primitive embeddings, where each point represents an action primitive segmented by our pipeline, embedded via the *Frozen Transformer* (Section 3.2.1), and colored according to its *k*-means-assigned cluster identity. The visualization reveals distinct, well-separated clusters that, upon manual inspection, correspond to semantically coherent groupings within the discrete and countable set of workstation tasks. As can be seen from the data, this effectively disambiguates coarse-grained action types such as Cluster 1: ‘Move Baffle’, Cluster 2: ‘Replace Motor’, and Cluster 3: ‘Grasp Motor’.

This analysis provides compelling qualitative evidence supporting our claim. The *Transformer* embeddings of the primitives form dense, well-separated clusters that *k*-means reliably identifies, demonstrating that our pipeline automatically discovers and organizes the inherent action vocabulary of the workstation in an unsupervised manner.

Table 3. Clustering results on the Exocentric View dataset (6,444 segments, $k = 3$), comparing our Frozen Transformer embedding with a strong non-temporal aggregation baseline.

Embedding Method	Silhouette	Calinski-Harabasz
Attention-Norm Pooling	0.498	3523.6
Ours (Frozen Transformer)	0.588	3919.2

Table 3 provides quantitative validation of our approach. The results clearly indicate that our *Frozen Transformer encoder* significantly outperforms the mean-pooling baseline across both unsupervised metrics. This improvement con-

firmes our hypothesis: explicitly modeling temporal dynamics through self-attention capabilities, which is unachievable by simple mean-pooling, is essential for distinguishing subtle action patterns that are obscured by naive aggregation strategies. The ability to capture fine-grained dependencies enables the formation of semantically coherent action clusters, which are critical for effective downstream learning tasks.

4.5. Semantic Coherence Validation via VLM

We now quantitatively validate the semantic coherence of the discovered clusters using our *Intra-Cluster Semantic Similarity (ICSS)* metric (Section 3.2.3). As shown in Table 4, we compare the average ICSS *within* each discovered cluster to a *random-pair* baseline. The results in Table 4

Table 4. VLM-based Semantic Coherence (ICSS): Mean intra-cluster similarity ($\pm$ std) for discovered clusters. The baseline samples random pairs from the entire dataset irrespective of clusters and thus, by definition, only provides a single *Overall* metric for comparison.

Metric	Baseline	Ours (<i>k</i> -means)
Cluster K_1	–	0.919 ± 0.040
Cluster K_2	–	0.929 ± 0.029
Cluster K_3	–	0.922 ± 0.036
Overall ICSS	0.804 ± 0.127	0.926 ± 0.033

show that the mean similarity *within* our discovered clusters (Overall ICSS: 0.926 ± 0.033) is substantially higher than the similarity of random pairs drawn from the entire dataset (Baseline: 0.804 ± 0.127). Furthermore, the tight standard deviations of our cluster scores, compared to the baseline’s much wider variance, provide strong quantitative evidence that our pipeline successfully discovers and groups semantically coherent action primitives. This automated semantic validation confirms the output is a structured, high-quality dataset of “countable” actions, perfectly suited for the downstream task of VLA pre-training.

4.6. Ablation Studies

We perform ablation studies to rigorously evaluate the impact of our key design choices within the pipeline. Table 5 summarizes these results. First, substituting our specialized motion tokenizer M_θ with generic CLIP features results in a severe degradation in both segmentation and clustering performance, confirming that a domain-specific motion tokenizer is critical for capturing the fine-grained nuances of industrial tasks. Second, the effectiveness of our E_{action} metric is contingent upon its computation in the *quantized* space (S_q); applying it to raw velocities or latent representations prior to quantization yields poor results. This finding validates the importance of discretization for accurately

capturing semantic intent. Finally, in the clustering stage, the Frozen Transformer encoder outperforms simple mean-pooling, demonstrating that explicitly modeling temporal dynamics is essential for discovering coherent and semantically meaningful action vocabularies.

Table 5. Ablation study on pipeline components, showing their impact on segmentation and clustering. Segmentation is evaluated using the strict **F1@2s** metric. Results are from the Exocentric View test set.

Configuration	F1@2s (%)	Cluster ICSS
Full Pipeline (Ours)	87.5	0.92
<i>Signal Source Ablation:</i>		
E_{action} from Pre-Quant. Latents	25.2	–
E_{action} from Raw Velocities	24.9	–
<i>Encoder Ablation:</i>		
w/o Transformer (Mean-pool)	–	0.84
<i>Representation Ablation:</i>		
w/o M_θ (e.g., CLIP)	27.2	0.75

5. Conclusion and Discussion

In this work, we addressed the critical data bottleneck for industrial VLA models by introducing Latent Action-based Primitive Segmentation (LAPS), the first unsupervised pipeline to discover finite sets of actions from passive video streams. Our core novelty is a segmentation paradigm shifted into an abstract latent action space, where our *Latent Action Energy* (E_{action}) metric robustly captures “behavioral intent” over physical movement. Validated on a real-world motor assembly dataset, LAPS significantly outperformed TAD baselines. Furthermore, our unsupervised discovery pipeline identified the finite action primitives, whose semantic coherence was quantitatively confirmed by our VLM-based ICSS metric. LAPS transforms raw observational data into a structured and learnable knowledge base, paving a scalable pathway for deploying embodied AI. At the current stage, our method is limited to highly repetitive tasks as they are found in the manufacturing domain.

In future work we want to elaborate how our pipeline can be extended towards other domains such as tasks in domestic households and hospitals. Furthermore, our next immediate step will be to bridge the gap from high-level task understanding towards task execution. To this end, we want to train a dual-arm manipulator to perform tasks from the manufacturing domain by teleoperation and to correlate those skills with our discovered latent space. Finally, to enable the transformation of structured latent knowledge into real-world task execution.

References

- [1] Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025. [1](#), [2](#)
- [2] Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Manuel Y Galliker, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization. In *9th Annual Conference on Robot Learning*, 2025. [1](#), [2](#)
- [3] Andrea Bonci, Pangcheng David Cen Cheng, Marina Indri, Giacomo Nabissi, and Fiorella Sibona. Human-robot perception in industrial environments: A survey. *Sensors*, 21(5):1571, 2021. [3](#)
- [4] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022. [2](#)
- [5] Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, Xuan Hu, Xu Huang, et al. Agibot world colosseum: A large-scale manipulation platform for scalable and intelligent embodied systems. *arXiv preprint arXiv:2503.06669*, 2025. [1](#), [2](#), [3](#)
- [6] Tadeusz Caliński and Jerzy Harabasz. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27, 1974. [6](#)
- [7] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. [7](#)
- [8] Haw-Shiuan Chang, Ruei-Yao Sun, Kathryn Ricci, and Andrew McCallum. Multi-clas bert: An efficient alternative to traditional ensembling. *arXiv preprint arXiv:2210.05043*, 2022. [5](#)
- [9] Y Chen, Y Ge, W Tang, Y Li, Y Ge, M Ding, Y Shan, and X Liu. Moto: Latent motion token as the bridging language for learning robot manipulation from videos, 2025. URL <https://arxiv.org/abs/2412.04445>. [1](#)
- [10] Jeremy A Collins, Loránd Cheng, Kunal Aneja, Albert Wilcox, Benjamin Joffe, and Animesh Garg. Amplify: Actionless motion priors for robot learning from videos. *arXiv preprint arXiv:2506.14198*, 2025. [2](#), [3](#), [5](#), [6](#)
- [11] Navneet Dalal, Bill Triggs, and Cordelia Schmid. Human detection using oriented histograms of flow and appearance. In *European conference on computer vision*, pages 428–441. Springer, 2006. [7](#)
- [12] Guodong Ding, Fadime Sener, and Angela Yao. Temporal action segmentation: An analysis of modern techniques. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(2):1011–1030, 2023. [2](#), [4](#)
- [13] Zexing Du, Xue Wang, Guoqing Zhou, and Qing Wang. Fast and unsupervised action boundary detection for action segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3323–3332, 2022. [2](#), [4](#), [6](#), [7](#)
- [14] Alireza Fathi, Xiaofeng Ren, and James M Rehg. Learning to recognize objects in egocentric activities. In *CVPR 2011*, pages 3281–3288. IEEE, 2011. [6](#), [7](#)
- [15] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18995–19012, 2022. [1](#), [2](#)
- [16] Nikita Karaev, Yuri Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. CoTracker3: Simpler and better point tracking by pseudo-labelling real videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6013–6022, 2025. [2](#), [3](#)
- [17] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024. [1](#), [2](#)
- [18] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024. [2](#)
- [19] Hilde Kuehne, Ali Arslan, and Thomas Serre. The language of actions: Recovering the syntax and semantics of goal-directed human activities. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 780–787, 2014. [6](#), [7](#)
- [20] Yuerong Li, Zhengrong Xue, and Huazhe Xu. Otas: unsupervised boundary detection for object-centric temporal action segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 6437–6446, 2024. [2](#), [6](#), [7](#)
- [21] Fabian Mentzer, David Minnen, Eirikur Agustsson, and Michael Tschannen. Finite scalar quantization: Vq-vae made simple. *arXiv preprint arXiv:2309.15505*, 2023. [3](#)
- [22] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979. [4](#)
- [23] Abby O’Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6892–6903. IEEE, 2024. [1](#), [2](#)
- [24] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. Pmlr, 2021. [5](#)

- [25] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987. [6](#)
- [26] Dominik Schmidt and Minqi Jiang. Learning to act without actions. *arXiv preprint arXiv:2312.10812*, 2023. [2](#)
- [27] Peiyao Wang, Yuewei Lin, Erik Blasch, Haibin Ling, et al. Efficient temporal action segmentation via boundary-aware query voting. *Advances in Neural Information Processing Systems*, 37:37765–37790, 2024. [2](#)
- [28] Angchi Xu and Wei-Shi Zheng. Efficient and effective weakly-supervised action segmentation via action-transition-aware boundary alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18253–18262, 2024. [2](#)
- [29] Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Sejune Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, et al. Latent action pretraining from videos. *arXiv preprint arXiv:2410.11758*, 2024. [1](#), [2](#)
- [30] Ziqian Zhong and Jacob Andreas. Algorithmic capabilities of random transformers. *Advances in Neural Information Processing Systems*, 37:104357–104382, 2024. [5](#)
- [31] Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pages 2165–2183. PMLR, 2023. [1](#), [2](#)